CLAIMS

What is claimed is:

1. A method for exporting a software model to different operating systems, the method comprising:

providing a software model;

providing a porting layer, the porting layer porting the software model to an operating environment;

providing an operating environment, the operating environment common to all the different operating systems; and

providing a plurality of operating system abstraction layers, each abstraction layer designed to abstract the operating environment to at least one targeted operating system.

2. The method of claim 1 wherein the at least one targeted operating system is a single operating system.

3. The method of claim 1 wherein the at least one targeted operating system is two operating systems and the method for exporting a software model in a wireless device, a first of the two operating systems is a system operating system and a second of the two operating systems is a communication operating system.

4. The method of claim 3 wherein the system operating system operates on an advanced reduced instruction set processor (RISC) and the communication operating system operates on a digital signal processor (DSP).

5. The method of claim 3 wherein a communication module facilitates communication between the RISC and DSP.

6. The method of claim 5 wherein the communication module has an associated shared memory for use in performing operations of code derived from the software model.

7. The method of claim 1 wherein the at least one target operating system is a plurality of operating systems.

8. The method of claim 1 wherein the operating environment operates independently of processor boundaries.

9. The method of claim 7 wherein the operating system abstraction layer defines the processor boundaries and facilitates communication across the processor boundaries.

10. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system, the operating system performing code generated from a software model, the software model being ported to an operating environment providing an operating environment, the operating environment common to all the different operating systems, an operating system abstraction layer abstracts the operating environment to each associated operating system.

11. The wireless communication device of claim 10 wherein the wireless communication device is a wireless transmit/receive unit.

12. The wireless communication device of claim 11 wherein the at least one system processor is a advanced reduced instruction set processor and the communication processor is a digital signal processor.

13. The wireless communication device of claim 10 wherein the operating environment operates independently of processor boundaries.

14. The wireless communication device of claim 13 wherein the operating system abstraction layer defines the processor boundaries and facilitates communication across the processor boundaries.

15. A method for porting software developed using a single threaded modeling tool to a multiple threaded environment, the method comprising:

using the single threaded modeling tool to model the software; and

providing a porting layer, the porting layer performing as follows:

porting in variables into a multiple threaded operating environment by reference and not as variables so that each thread can access variables by reference.

16. The method of claim 15 wherein the single threaded modeling tool produces variables as global variables and not using the global variables in operation of a plurality of threads in the multiple threaded operating environment.

17. The method of claim 15 wherein the porting layer comprises a root process table having process description block entries, each process in the process description block entry having static variables.

18. The method of claim 15 wherein a modeling language used in the software development is SDL and the single threaded software tool is Telelogic Tau C-micro with light integration.

19. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system, the operating system performing code generated from a software model, the software model developed using a single threaded modeling tool, a porting layer ports code generated by the single threaded modeling tool to a multiple threaded environment, the porting layer porting in variables into the multiple threaded operating environment by reference and not as variables so that each thread can access variables by reference.

20. The wireless communication device of claim 19 wherein the single threaded modeling tool produces variables as global variables and not using the global variables in operation of a plurality of threads in the multiple threaded operating environment.

21. The wireless communication device of claim 19 wherein the porting layer comprises a root process table having process description block entries, each process in the process description block entry having static variables.

22. The wireless communication device of claim 19 wherein a modeling language used in the software development is SDL and the single threaded software tool is Telelogic Tau C-micro with light integration.

23. The wireless communication device of claim 19 wherein the wireless communication device is a wireless transmit/receive unit.

24. A method for synchronizing a plurality or threads in a software environment, each thread being an independent path of execution, the method comprising:

for a plurality of threads in a thread group requiring synchronization, initiating each of the plurality of threads;

awaiting an initiation return from all the threads in the thread group indicating that the initiating of that thread is complete; and

starting execution of all the threads in the thread group after the awaiting.

25. The method of claim 24 further comprising providing a plurality of synch threads, each synch thread initiating a thread in the group and starting execution of that thread.

26. The method of claim 25 further comprising providing a synch thread data table, the synch thread data table having a priority for each thread, an initialization function for each thread, an initialization argument for each thread, a reference to a main loop function for each thread and a main loop argument passed to be passed to each thread.

27. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system, software operating on the operating system synchronizes a plurality of threads within a thread group in a software environment, each thread being an independent path of execution, for the plurality of threads in the thread group requiring synchronization, each of the plurality of threads is initiated; an initiation return is awaited from all the threads in the thread group indicating that the initiating of that thread is complete; and execution of all the threads in the thread group is started after the awaiting.

28. The wireless communication device of claim 27 wherein the software further operates by providing a plurality of synch threads, each synch thread initiating a thread in the group and starting execution of that thread.

29. The wireless communication device of claim 27 wherein the software further operates by providing a synch thread data table, the synch thread data table having a priority for each thread, an initialization function for each thread, an initialization argument for each thread, a reference to a main loop function for each thread and a main loop argument passed to be passed to each thread.

30. The wireless communication device of claim 27 wherein the wireless communication device is a wireless transmit/receive unit.

31. An operating system abstraction layer comprising:

an interface with an operating environment, the operating environment operating independent of underlying operating systems;

an operating system independent module for performing operations that are not related to a target operating system;

an operating system dependent module for performing operations that are related to the target operating system; and

an interface with the target operating system.

32. A method for abstracting an operating environment to a plurality of operating systems, the method comprising:

providing an operating environment, the operating environment common to all the different operating systems; and

providing a plurality of operating system abstraction layers, each abstraction layer designed to abstract the operating environment to at least one targeted operating system.

33. The method of claim 32 wherein each abstraction layer has a same operating system dependent module and a different operating system independent module.

34. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system, the operating system performing code from an operating system abstraction layer, the abstraction layer interfacing with the operating environment and having an operating system independent module for performing operations that are not related to a target operating system and an operating system dependent module for performing operations that are related to the target operating system.

35. The wireless communication device of claim 34 wherein the wireless communication device is a wireless transmit/receive unit.

36. A method for controlling software timing in a multiple operating system environment, the method comprising:

providing a time manager;

operating system constructs in each of a plurality of operating systems sending callbacks to the time manger;

after a time period specified by each call back, the time manager sending a callback message to the construct of that callback; and

the construct operating in response to the received message.

37. The method of claim 36 wherein the time manager has an associated callback list, the associated call back list is an ordered list of callback time requests of the sent callbacks.

38. The method of claim 36 wherein the time manager has a timer, the timer has a plurality of threads that are waken up in response to expiration of callback time requests.

39. The method of claim 36 wherein the time manager has a timer, the timer scheduling a shortest callback delay and after expiration of the shortest callback delay, scheduling a next shortest callback delay.

40. A time manager for controlling software timing in a multiple operating system environment, the time manager comprising:

an input configured to receive callbacks sent from operating system constructs in each of a plurality of operating systems;

a timer for after a time period specified by each call back, producing a thread to indicate expiration of the callback to the construct of that callback.

41. The time manager of claim 40 wherein the timer scheduling a shortest callback delay and after expiration of the shortest callback delay, scheduling a next shortest callback delay.

42. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system; and

a time manager for controlling software timing in each operating system, the time manager comprising:

an input configured to receive callbacks sent from operating system constructs in each of the plurality of operating systems;

a timer for after a time period specified by each call back, producing a thread to indicate expiration of the callback to the construct of that callback.

43. The wireless communication device of claim 42 wherein the timer scheduling a shortest callback delay and after expiration of the shortest callback delay, scheduling a next shortest callback delay.

44. The wireless communication device of claim 42 wherein the wireless communication device is a wireless transmit/receive unit.

45. A method for software processes to communicate across processor boundaries, the method comprising:

providing a local process associated with a local processor and a remote process associated with a remote processor:

providing a local queue for the local process;

providing or creating a remote queue for the remote process;

the local process having a put request in the local queue indicating a message to be sent to the remote process; and

putting the message in the remote queue.

46. A wireless communication device comprising:

at least one system processor and at least one communication processor;

a communication module to facilitate communication between each system and communication processor;

a shared memory associated with the communication module;

each system processor and communication processor having an associated operating system;

a local process is associated with a local processor of the system and communication processors and a remote process associated with a remote processor of the system and communication processors;

a local queue is associated with the local process; and

a remote queue is associated with the remote process; and

wherein the local process having a put request in the local queue indicating a message to be sent to the remote process; and putting the message in the remote queue.

47. The wireless communication device of claim 46 wherein the wireless communication device is a wireless transmit/receive unit.